

Phoenix Overview

MSR - VC – CLR

Feb 27, 2003

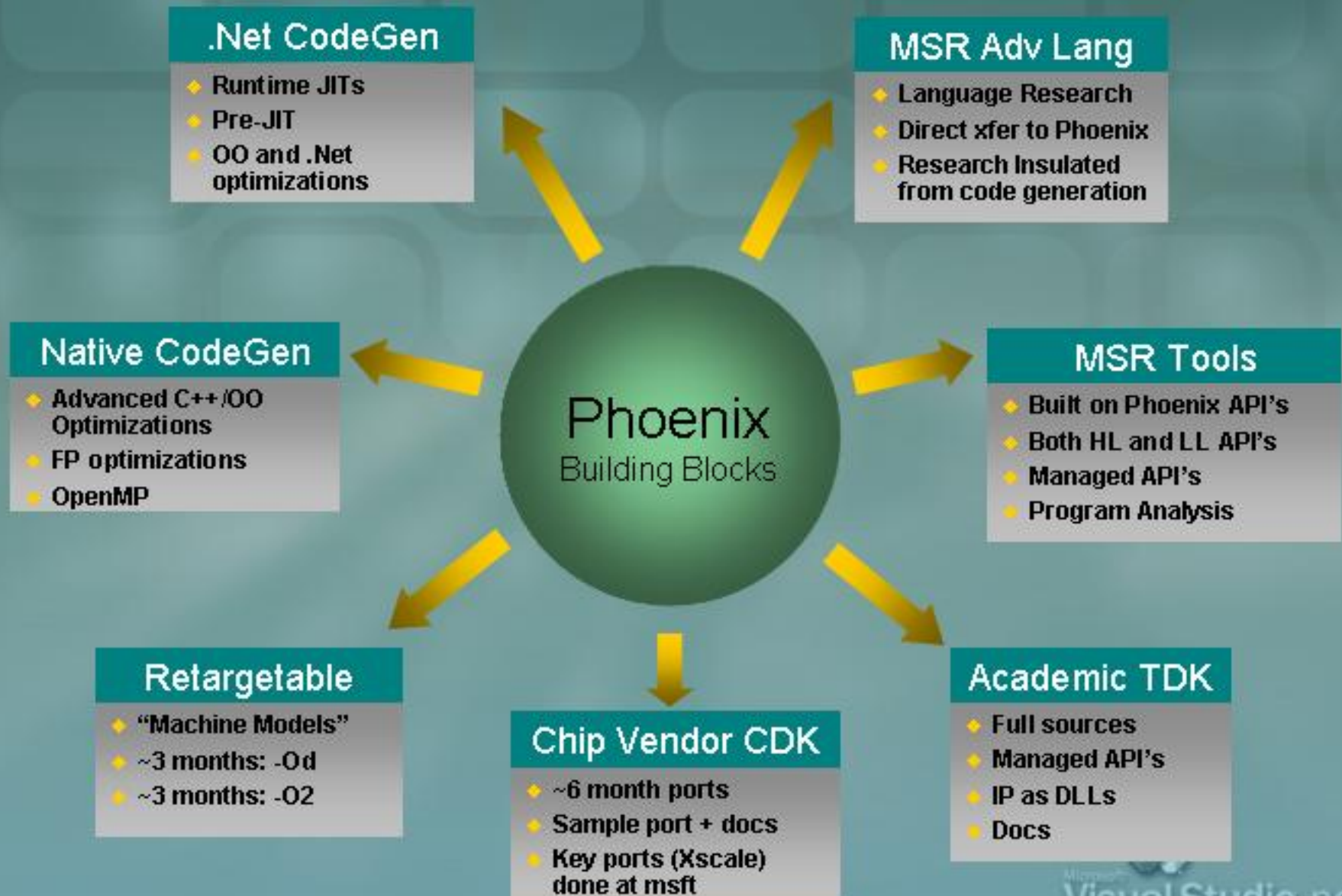
What is Phoenix?

**The Next Generation Platform
for building retargetable,
optimizing JITers, Compilers
and Program Analysis Tools
for use in Product and
Research groups**

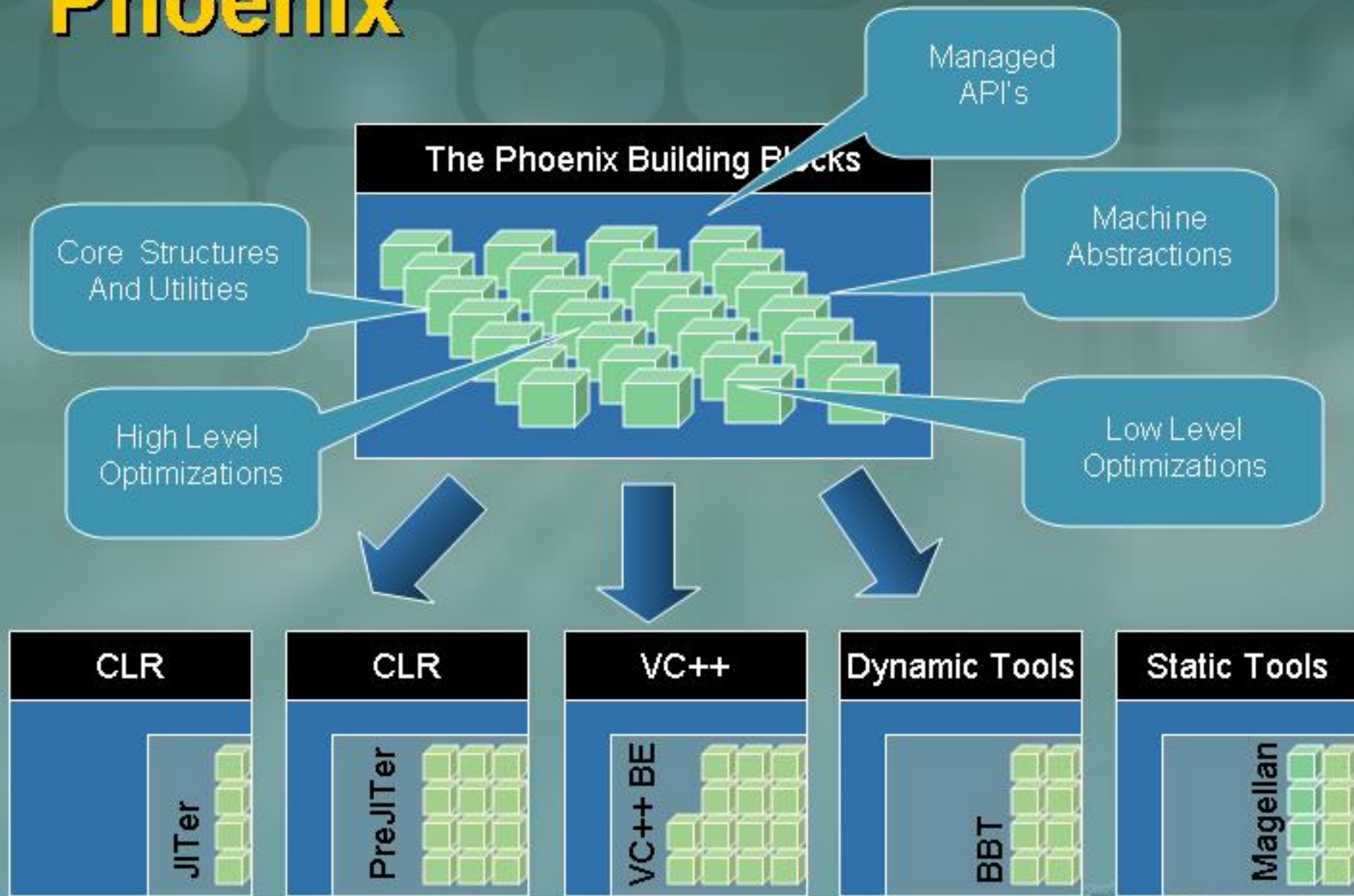
The situation today

- ◆ 20+ code generators at msft alone!
- ◆ It's not an x86-only world any more
- ◆ High cost of technology xfer from MSR
- ◆ .Net paradigm shift
- ◆ Minimal Academic / Community presence
- ◆ Less than ideal solution for WinCE chip vendors

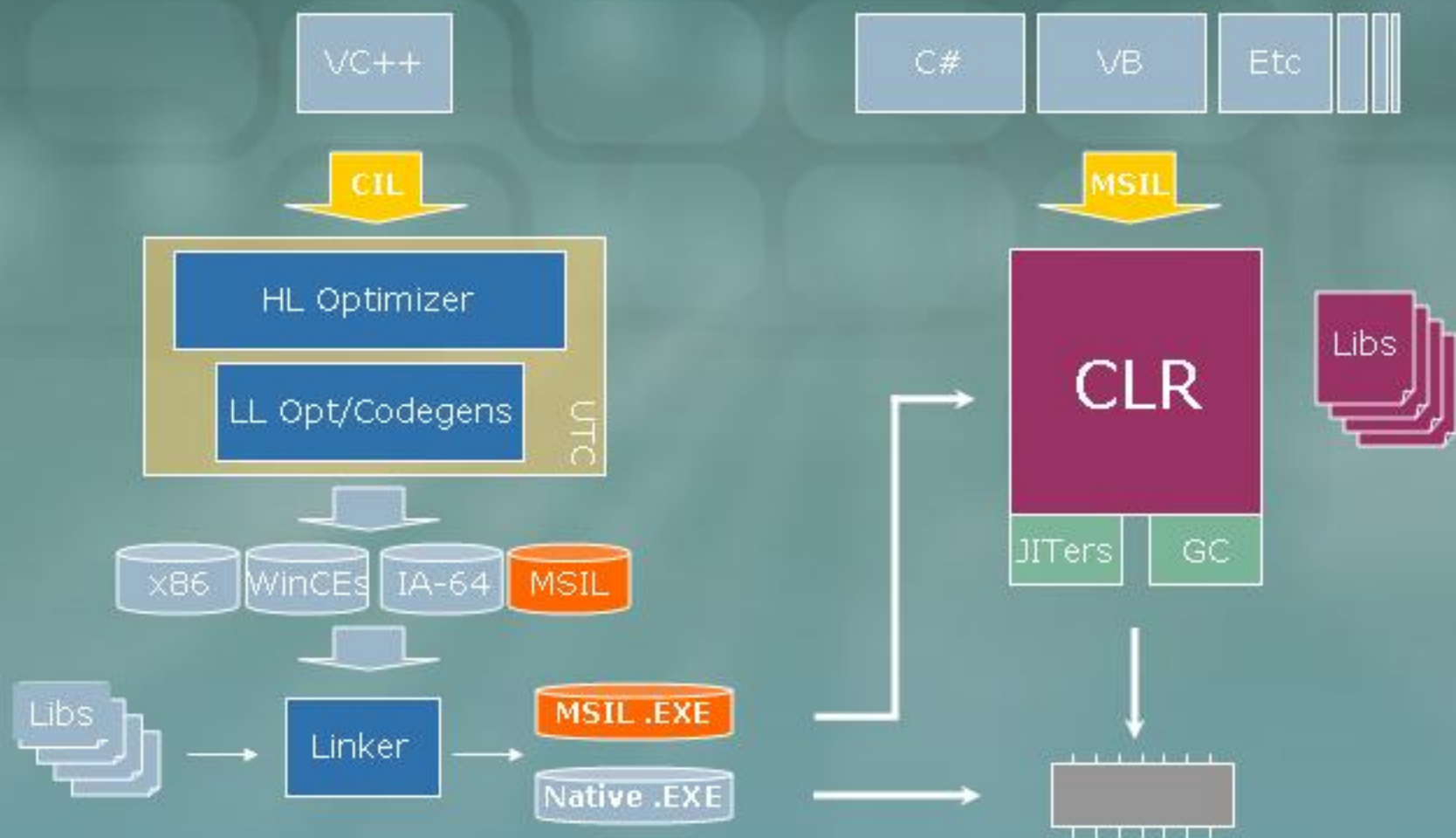
How will Phoenix help?



Phoenix

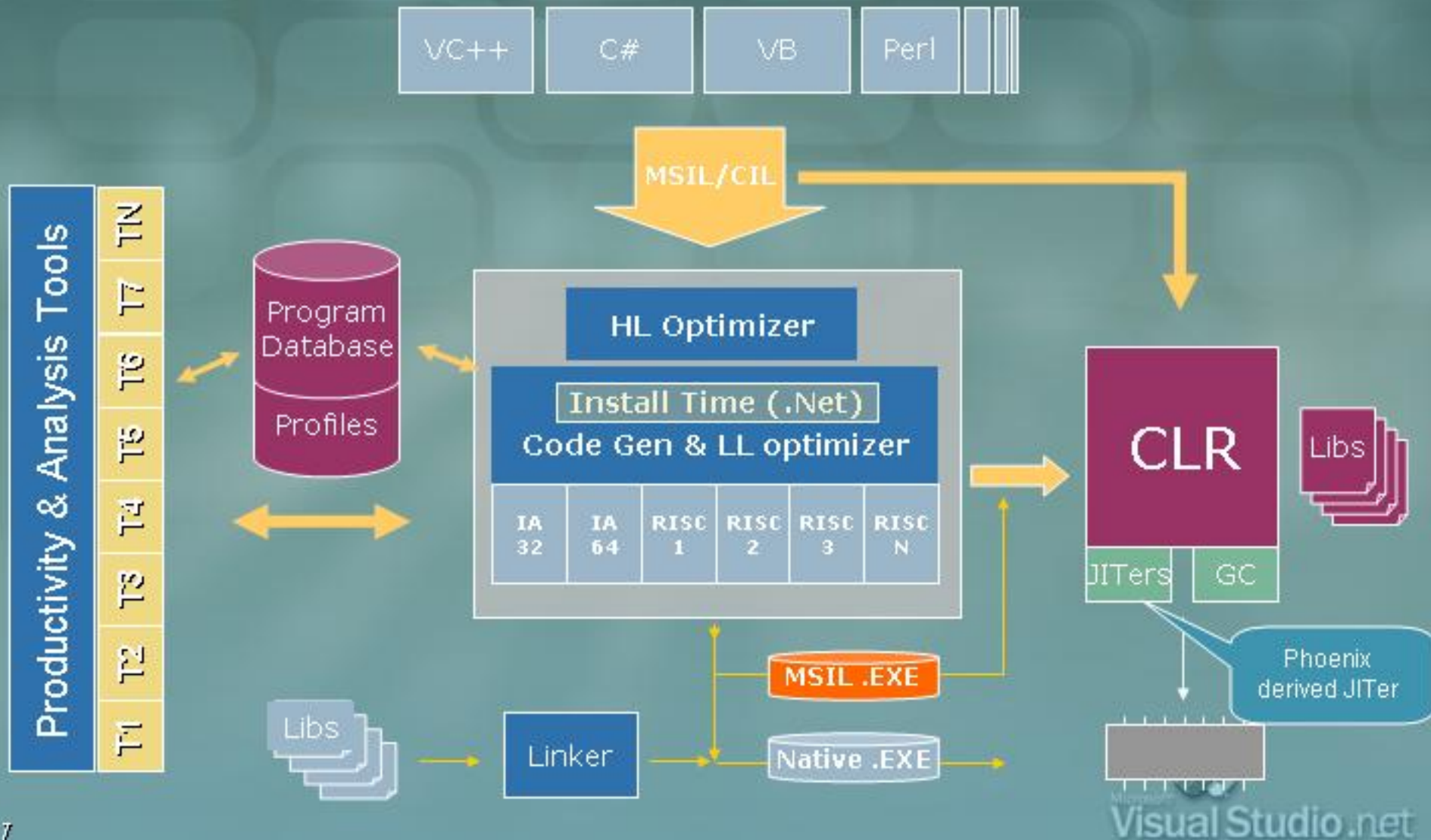


Compiling Native and .Net code today

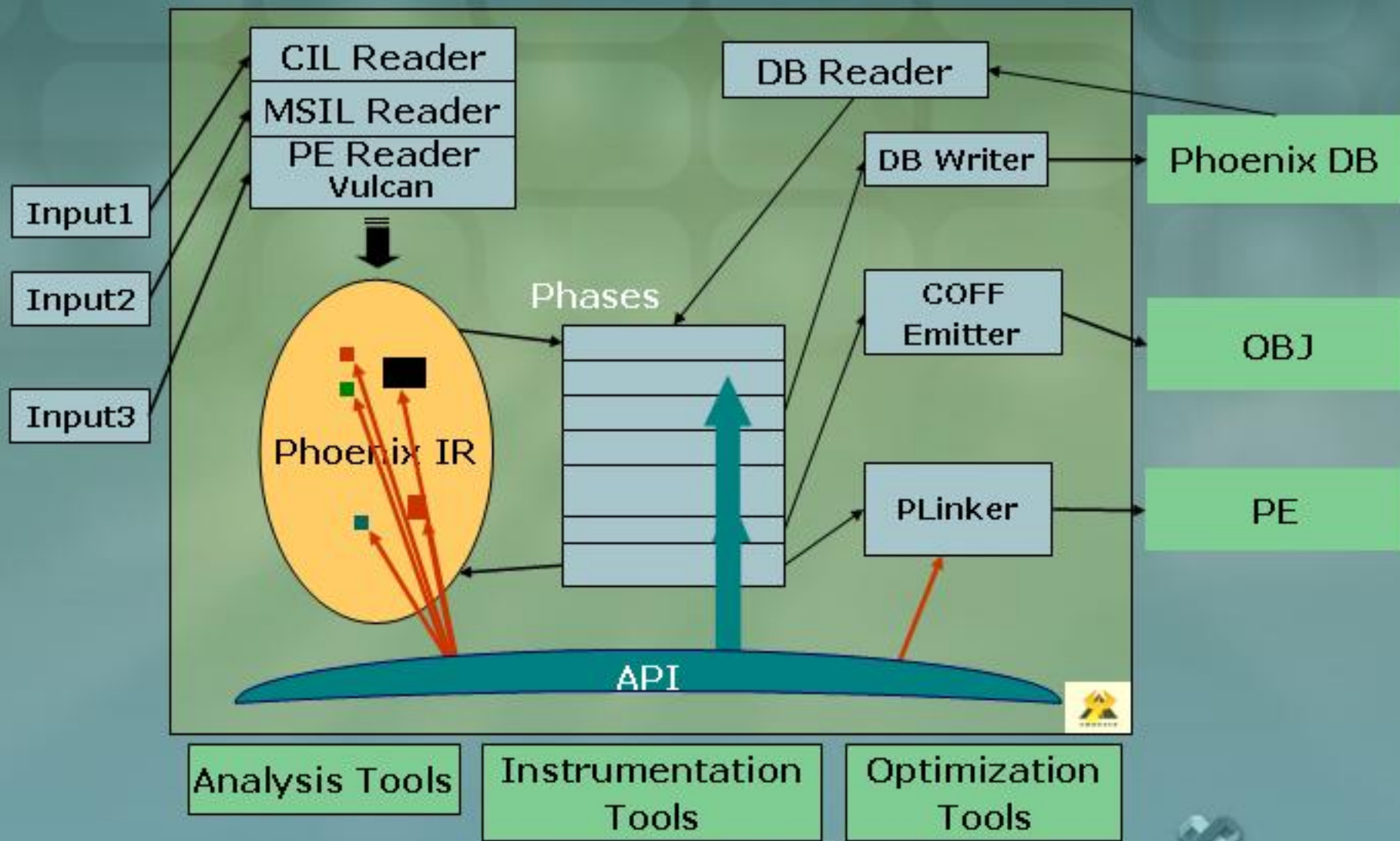


What's wrong with this picture?

Compilation model for future



Phoenix Infrastructure



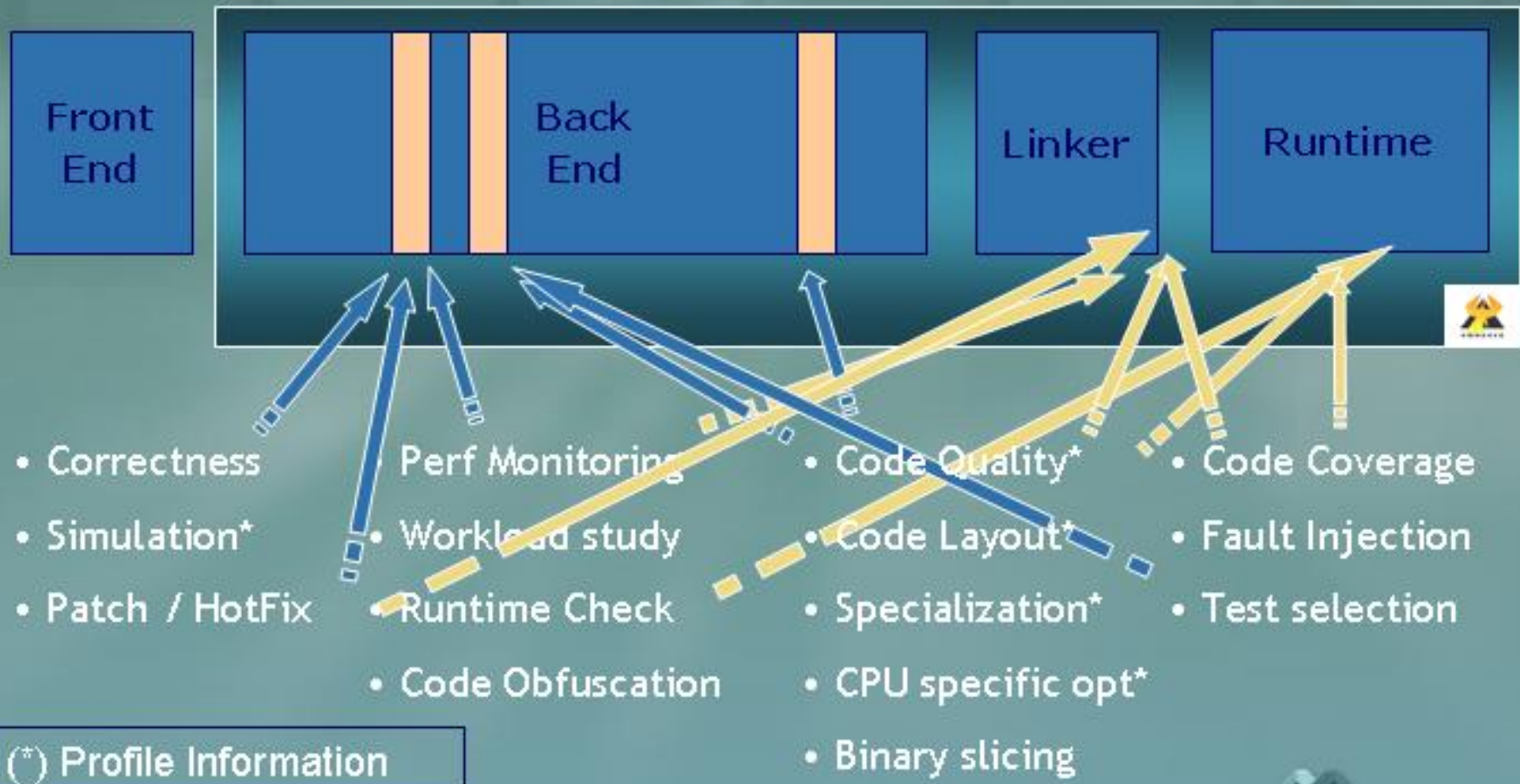
Phoenix Tools Opportunities

Static Analysis

Instrumentation

Optimization

Testing



Phoenix as a Tools Platform

- ◆ Rich set of managed APIs
- ◆ Unprecedented access to compiler internals
- ◆ Deep integration between analysis tools and compilers
 - ◆ An industry 1st
- ◆ Research and Product tools platform
- ◆ All MSR current & future tools will be hosted on Phoenix (mgd)

Highly integrated compiler+tools platform = competitive advantage

Scenarios for Phoenix V1

Orcas timeframe

- ◆ Highly optimized .Net code
 - ◆ “PreJIT Centric”
 - ◆ Profile driven
- ◆ Highly optimized Native code
 - ◆ Int & FP
- ◆ Wins in RWC, SPEC, .Net benchmarks
- ◆ Highly integrated MSR tools
- ◆ Academic research initiatives
- ◆ Potential solution for longer lead projects (eg Xbox)
- ◆ WinCE CDK

X86 is
1st target

Progress: Milestones

- ◆ **M0 (end = May 30 2002)**
 - ◇ Design work
 - ◇ Implementation language investigations
 - ◇ Build up team
- ◆ **M1 (end = Jul 30 2002)**
 - ◇ Core Data Structures, API's, IR, etc.
 - ◇ Read & dump IL for a synthesized test
- ◆ **M2 (end = Sep 30 2002)**
 - ◇ MSIL & CIL Readers
 - ◇ Read & dump x86 for "hello world" – no exec
 - ◇ PE Reader & Bartok hook ups

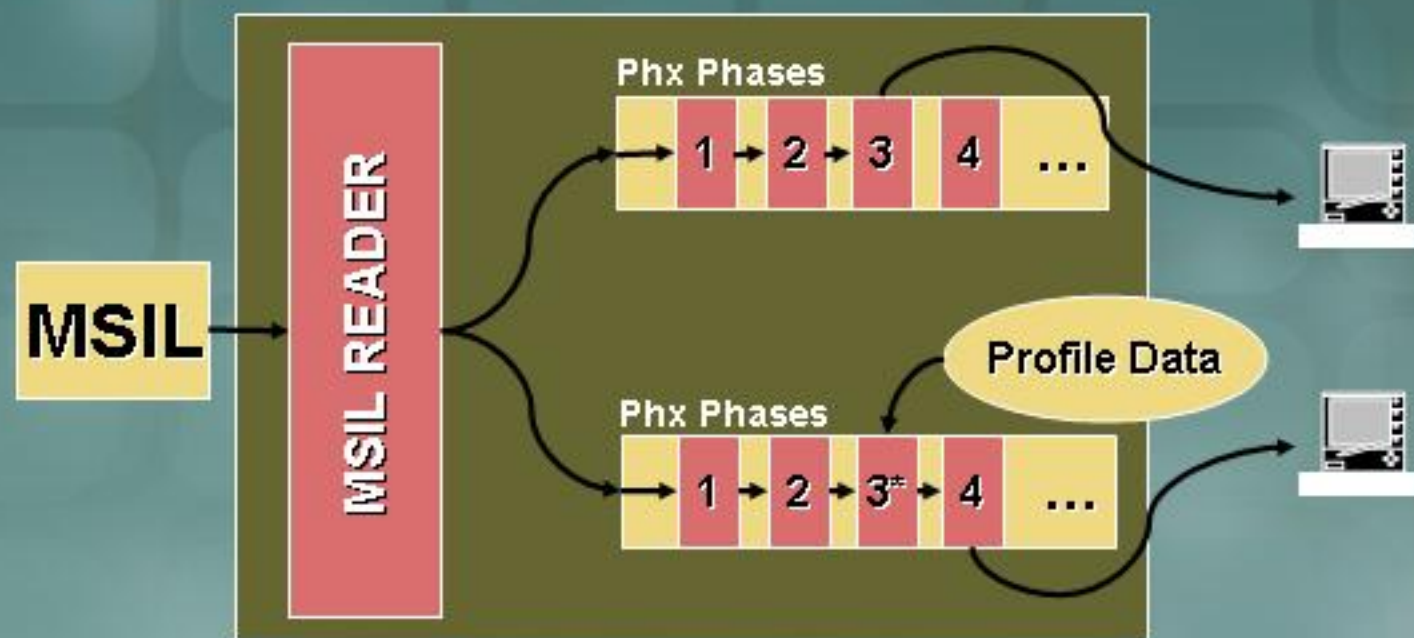
Progress: Milestones cont.

- ◆ M3 (end = Nov 29 2002)
 - ◇ Read MSIL, CIL, X86, Convert to IR and execute code
 - ◇ Productivity tool hookups
 - ◇ Bartok research compiler hookup
 - ◇ Machine models for IA32, IA64, XScale completed

Phoenix Compiler Demos

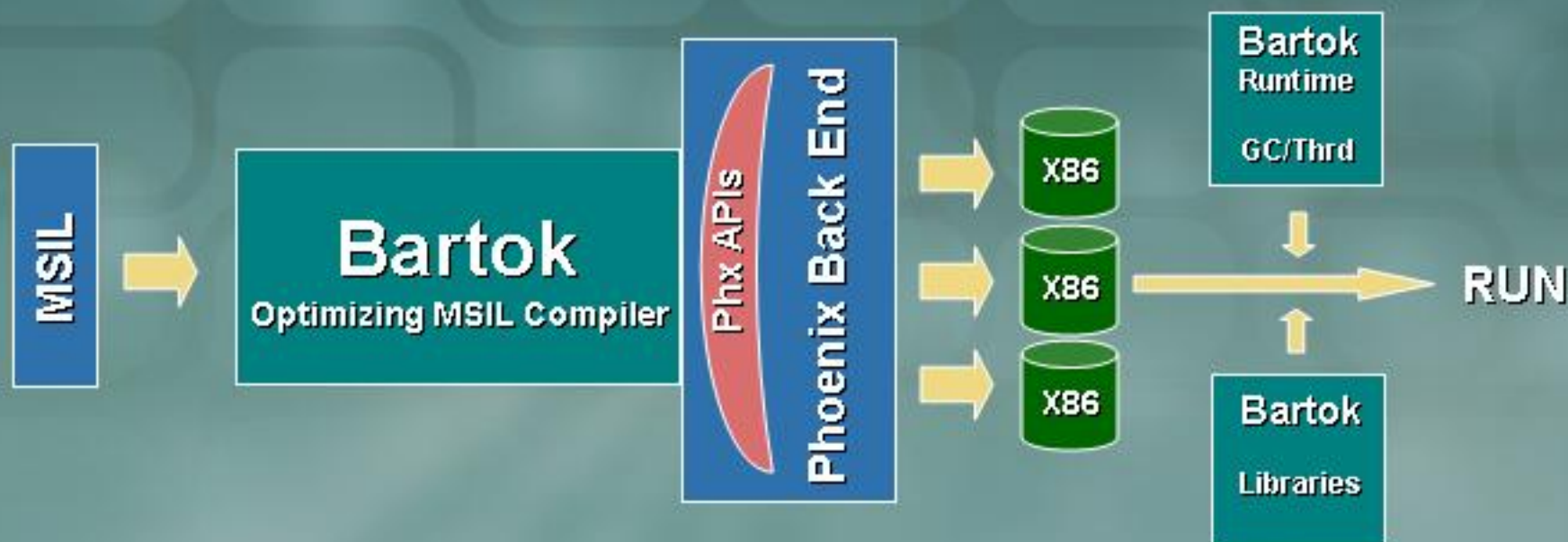
- ◆ NGEN (pre-JIT) on X86
- ◆ Native C++ Backend on IA64
- ◆ JIT on X86

Demo2 - Visualizer



- ◆ IR / Instruction Visualizer for Phx
 - ◆ Tool *drives and controls* the compilation process
 - ◆ Used for program comprehension; analysis; debugging
 - ◆ Compare IR pre & post transformation after each phase
 - ◆ Single snap shots and multi phase visual “diffs”
 - ◆ Dynamic – can edit graph and continue compilation

Demo3 – Bartok Compiler



◆ Optimizing Ahead-of-time MSIL Compiler

- ◆ Research vehicle for .Net optimizations
- ◆ Has its own mini-runtime: GC, threads, ...
- ◆ Will be using Phx as its code generator
- ◆ Optimization work will flow directly into Phx
- ◆ Can build itself on top of Phx already!

Phoenix In Research

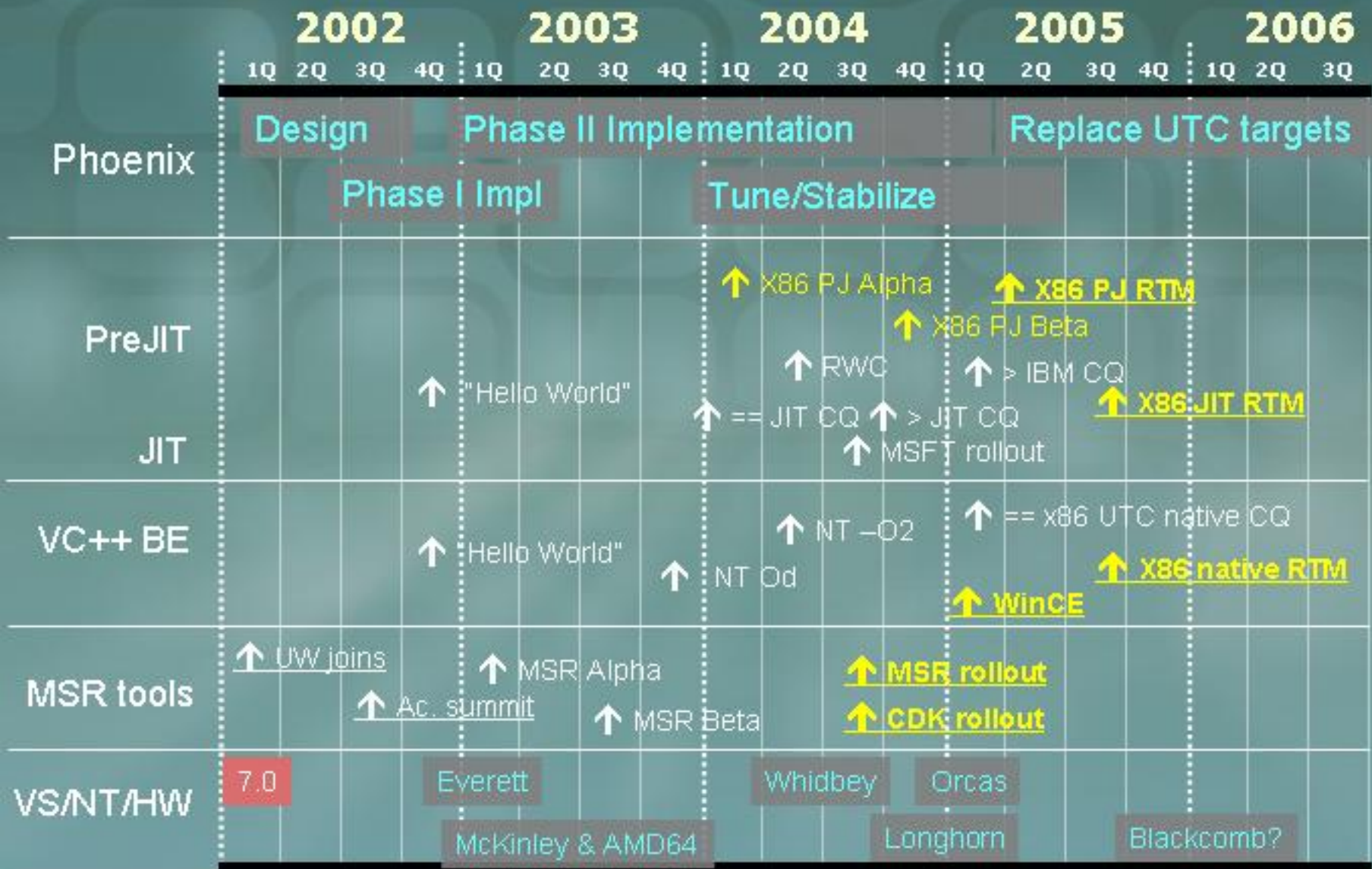
Academic Engagement

- ◆ **Win the hearts and minds of the compiler/tool academics**
 - ◆ Need CLR src access for .Net research work
- ◆ **Phoenix available as shared-source**
 - ◆ IP protected portions released in binary form
- ◆ **Recruit 3-4 researchers from domestic universities to be early adopters of Phoenix in their research**
 - ◆ Fund research
 - ◆ Hire students as interns in MSR/Phoenix group
 - ◆ Produce publishable results using Phoenix

2003

- ◆ **M4 (end = April 03)**
 - ◆ Further design work + code cleanup
 - ◆ Key optimizations start
 - ◆ GC, EH, etc.
 - ◆ Patents, IP related work (~ 6)
 - ◆ Industry & Academic partners
 - ◆ Additional MSR tools
 - ◆ Technology transition planning
- ◆ **M5 (end = Aug 03)**
 - ◆ Core optimizations in place
 - ◆ Large test cases: NT, SQL, Office
 - ◆ Internal Alpha

Time line



Schedule biased towards delivering an X86 PreJIT



Summary

- ◆ This is an ambitious project and critical to the future of the .Net platform
- ◆ Phoenix can give us a tangible competitive advantage
- ◆ Phoenix will be a great Research Platform
- ◆ Great collaboration between VC, MSR and CLR!